Residual Capsule Network

Sree Bala Shruthi Bhamidi
Electrical and Computer Engineering
Purdue School of Engineering, IUPUI
Indianapolis, USA
sbhamidi@iupui.edu

Mohamed El-Sharkawy
Electrical and Computer Engineering
Purdue School of Engineering, IUPUI
Indianapolis, USA
melshark@iupui.edu

***Abstract*—Convolution Neural Network (CNN) has been the most influential innovations in the filed of Computer Vision. CNN have shown a substantial improvement in the field of Machine Learning. But they do come with their own set of drawbacks - CNN need a large dataset, hyperparameter tuning is non-trivial and importantly, they lose all the internal information about pose and transformation to pooling. Capsule Networks have addressed the limitations of CNNs and have shown a great improvement by calculating the pose and transformation of the image. On the other hand, deeper networks are more powerful than shallow networks but at the same time, more difficult to train. Simply adding layers to make the network deep has led to vanishing gradient problem. Residual Networks introduce skip connections to ease the training and have shown evidence that they can give good accuracy with considerable depth. Putting the best of Capsule Network and Residual Network together, we present Residual Capsule Network, a framework that uses the best features of both Residual and Capsule Networks. In the proposed model, the conventional Convolutional layer in Capsule Network is replaced by skip connections like the Residual Networks to decrease the complexity of the Baseline Capsule Network and seven ensemble Capsule Network. We trained our model on MNIST and CIFAR-10 datasets and have noted a significant decrease in the number of parameters when compared to the Baseline models.**

***Index Terms*—Convolution Neural Network, Computer Vision, Deep Learning, Capsule Network, Residual Network, Dynamic Routing, BlueBox 2.0**

I. Introduction

Convolutional Neural Networks (CNNs) have become an integral part of machine learning. Even with a history of more than 20 years, CNNs have shown that there is always a room for improvement. Since concept of Deep Convolutional Network [1] has been rolled out, we can see a significant improvement in challenging tasks like Image Classification, Image Recognition. Though the deeper networks did improve the performance of the neural network models, stacking of layers brought in a new problem of vanishing gradients. This problem has been alleviated with the introduction of a new network Residual Network (ResNet) [2]. The network adds Skip connections between the layers i.e., the outputs of the previous layers are added to the outputs of the stacked layers in a feedforward manner. As a matter of fact, ResNets were not the first to add skip connections in the network. Highway Networks [14] were the first of architectures to successfully train deep networks with large number of layers. Highway Networks with hundreds of layers can be optimized and trained effortlessly using gating units. These gates control the flow of amount of information across the connection. However, Highway Networks did not outperform Residual Networks. Adding the skip connections not only decrease the number of parameters, but also helps in concatenating the feature maps for a better gradient flow across deeper networks.

Convolutional Neural Networks are our go-to algorithm when it comes to object recognition or object detection. But there are many things that are very unlike the brain that are making the CNNs work not as well as they could. One thing that is missing in Neural Networks is the notion of entity. Sabour et al. [3] pointed out the drawbacks of the traditional CNNs. Convolutional Neural Networks use multi layers of feature detectors which are replicated across space. These feature extractors with subsampling pooling layers attend only to the active features. In other words, pooling gives only a small amount of translational invariance at each level that is, the exact location of the most active feature extractor is ignored. Pooling also reduces the number of inputs to the next layer of the feature extractor. The downside to pooling is that they fail to use an underlying linear manifold which would deal easily with the effects of viewpoint. We do not want the neural activities to be invariant of the viewpoint instead we want the knowledge of the viewpoint which can be applied to a new viewpoint. Convolutional Neural Networks try to make the neutral activities invariant to small changes in viewpoint by combining the activities of the pool. But it is better to aim at equivariance where the changes in viewpoint correspond to neural activities.

Sabor et al. [3] tossed the idea of nesting the layers instead of stacking them. The nested layer is called the capsule, which is a group of neurons. This model is robust to transformations in terms of rotations. The capsule network has two key features: layer based squashing and dynamic routing. It replaces the scalar-output feature detectors of CNNs with vector-output capsules and max-pooling with routing-by-agreement to achieve the state-of-the-art accuracy on the MNIST dataset. The authors have used just a single layer of convolution and capsules. Increasing the complexity and adding depth to the network is a possible improvement. On that basis, ResNets accelerate the speed of training of the deep networks. They reduce the vanishing gradient effect by increasing the depth of network instead of the width, resulting

This is the author's manuscript of the article published in final edited form as:
Bhamidi, S. B. S., & El-Sharkawy, M. (2019). Residual Capsule Network. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), 0557–0560. https://doi.org/10.1109/UEMCON47517.2019.8993019

in lesser parameters and obtaining higher accuracy in network performance.

Following the same instinct, we have proposed the use of Residual Network to increase the depth of Capsule Network. The ResNet [2] will be the input to the dynamic routing algorithm [3]. The proposed architecture not just shows a reduction in the model size but also reduces the inference time of the model. The method has been evaluated on MNIST and CIFAR-10 datasets and the results are compared to the Capsule Network keeping the parameters such as learning rate, learning decay, number of capsule parameters same as the model proposed by Sabor et. al [3].

II. Background

The study of neural networks and architectures has been dominant part of machine learning right from the start. But the recent fame of the neural networks has added fuel to the research in this domain. The different CNN architectures proposed have added more layers significantly increasing the parameters and the computational time. The lower layers detected basic features while the higher layers detected more complex features in addition to the lower layers. Though these structures of have boosted the performance, there was a vast increase in the number of parameters.

Highway Networks [4] were the first of architectures to successfully train deep networks with large number of layers. Highway Networks with hundreds of layers can be optimized and trained effortlessly using gating units. By introducing skip connections which are used as bypassing paths, ResNets [2] have achieved a striking performance on Image classification and Image recognition tasks.

The Capsule Networks [3] are a great leap into the neural networks. Capsules are a group of neurons that represent various properties of entities in an image. These properties may include parameters like colour, positions, size, orientation, hue, etc. Capsules output a vector, which implies that the lower level capsules selectively agree on the parent capsule. The connection strength between the lower level capsule and parent capsule is increased when the prediction for parent capsule matches with the actual output of the parent capsule. During training, all activity vectors are masked but the correct activity vector which is then used to reconstruct the input image. The output is used to compute the loss. This way, the network is stimulated to learn more representations of the image.

III. Methodology

Multiple layers give the networks a compelling advantage in learning to solve complex problems. However, increasing the depth has led to vanishing gradient problem which was addressed by ResNet [3] by adding skip connections. Adding these skip connections has reduced the number of parameters compared to the conventional CNN. Conceptually, a CNN model uses many layers and neurons in it to capture the different feature variants. On the other hand, Capsule network shares the same capsule across the network to detect different variants. We start with the Capsule Network [3] baseline model and add layers to form a deeper architecture. We then explore the effect on the performance of the tailored model.

A. Residual Capsule Network

The initial convolutional layer of the baseline Capsule Network [3] just converts the pixel intensities to vector activities of local feature detectors which are given as input to the primary capsules. But, for complex datasets this might not be enough to process further in the capsules. Hence, we try to increase the depth by adding more convolutional layers. We have replaced this convolutional layer with a deeper Residual Network architecture. The baseline network was modified to include eight layers of Residual Network [2] based on the skip connections. Each of these layers is added up to the final convolutional layer. The feature maps obtained as a result is fed in as an input to the primary capsules. Each of the convolutional layer in the ResNet block creates 32 feature maps. The main idea of Hinton et. al [3] is to not use max pooling or average pooling but to shift the focus on equivariance instead of invariance.

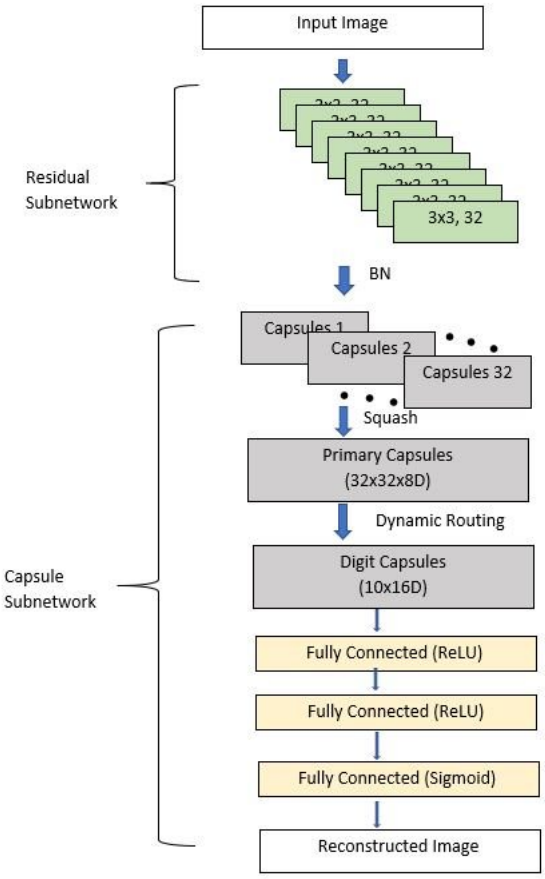

Fig. 1. Architecture of Residual Capsule Network

In this case, all the features detected at different levels of complexity of the Residual Network are combined and given as an input to the primary capsules. The primary capsule layer

is followed by the Digit Capsule layer where each of the 8D input vector gets a weight matrix and the 8D input space is converted to 16D capsule output space. The feature maps are then passed into the squash activation layer followed by the dynamic routing (or routing-by-agreement) algorithm. The matrices for each capsule and the coefficients from the Digit Capsule layer are used in the dynamic routing.

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{\boldsymbol{u}}_{j|i}, r, l$)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ softmax computes Eq. 3
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ squash computes Eq. 1
7: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
return $\mathbf{v}_j$

Fig. 2. Dynamic Routing Algorithm [3]

Routing a capsule to the capsule in the layer above based on relevancy is called Routing-by-agreement. Dynamic routing groups of capsules to form a parent capsule, and it calculates the capsules output. In dynamic routing we transform the vectors of an input capsule with a transformation matrix to form a vote, and group capsules with similar votes. If the activity vector has close similarity with the prediction vector, we conclude that both capsules are highly related [13]. Those votes eventually become the output vector of the parent capsule. The idea of squash function [3] [13] was that the length gives the probability of existence. The purpose is to output a number between 0 and 1, where the length of the input decides the probability.

The decoder then takes the 16D vector from the correct DigitCap and learns to decode it into an image of a digit. Decoder forces the capsules to learn features that are useful for reconstructing the original image. During training, all the activity vectors are masked except the right one. The input image is reconstructed using this activity vector. The output from the digit capsule layer is fed into decoder with 3 fully connected layers that is patterned into pixel intensities for the (input) image.

IV. Evaluation

We have evaluated our proposed model on the two basic datasets: MNIST, CIFAR-10 and compared the results with the Baseline Capsule Network by Sabor et. al [3], DCNET and DCNET++ by Phaye [8]. We ran all our evaluations on Aorus GeForce RTX 2080Ti GPU. We have run our models for 50, 120 epochs. Our implementation is in Keras and we use Adam optimizer with parameters set to 0.001 as initial learning rate and a decay rate of 0.9. We have used publicly available code [10] and made changes to suit our requirements for the proposed network. We have kept most of the parameters of the proposed Residual Capsule Network similar to the traditional Capsule Network, for fair comparisons.

A. MNIST Handwritten digits database

The MNIST database of handwritten digits [11], has a training set of 60,000 images and the testing set has a set of 10,000 images of each 28x28 in size.

TABLE I
PERFORMANCE OF VARIOUS CAPSULE NETWORK MODELS ON MNIST DATASET

Model	***Parameters***	***Test Accuracy****
Baseline Capsule Network	8.2M	99.67%
DCNet	11.8M	99.75%
DC++Net	10.5M	99.69%
Proposed Residual Capsule Network	7.7M	99.66%

$*Models are run for 50 Epochs each.$

The performance of various Capsule network models has been compared with the proposed models in Table 1.When the proposed Residual Capsule Network is compared with the Baseline Capsule Network model, the number of parameters decreased by 0.5M. But when compared to the DCNet, the parameters decreased by 4.1M on the MNIST dataset as shown in Table 1.

B. CIFAR-10 Dataset

CIFAR-10 [12] dataset consists of 60,000 images of 32x32 size each in 10 classes, with 6000 images per class. The images are divided into 50,000 training images and 10,000 test images. We compare our proposed Residual Capsule Network with seven ensemble Capsule Network [3] and DC++Net [8].

TABLE II
PERFORMANCE OF VARIOUS CAPSULE NETWORK MODELS ON CIFAR-10 DATASET

Model	***Parameters***	***Test Accuracy***
Baseline Capsule Network	7x14.5M = 101.5M	89.40% (7 model ensemble)
DCNet	11.88M	82.63%
DC++Net	13.4M	89.71% (120E)
Proposed Residual Capsule Network	11.86M	84.16%

When the number of parameters is compared with the seven-ensemble Capsule Network, the proposed network has fewer parameters by 89.64M. And when compared with the DC++Net, the proposed network has reduced 1.54M parameters as shown in Table 2.

V. Conclusion

The purpose of this paper is to re-design the Capsule Network to make it deep and simultaneously decrease the number of parameters compared to the baseline model. The proposed architecture uses the best of features of both Residual Network and Capsule Network to make the Capsule Network deeper yet efficient. To achieve this we decrease the complexity of the baseline Capsule Network by decreasing the number of parameters. We have replaced traditional Convolutional Layer

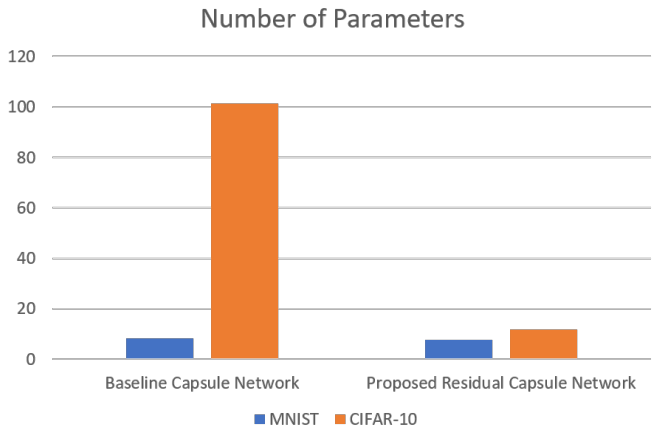

Fig. 3. Comparison of Number of Parameters

in the Capsule Network has been replaced by Residual (block-2) Network [2] to help decrease the model size proposed by Sabor et. al [3]. The proposed Residual Capsule Network model is potent of reducing the number of parameters by 6.09% by compromising on the accuracy by 0.01% when compared to Baseline Capsule Network and 34.74% reduction on architecture built by Dense Convolutional Network at the cost of 0.09% accuracy on MNIST dataset. On the other hand, the proposed Residual Capsule Network is capable of reducing the number of parameters by 88.32% when compared to the seven-ensemble Capsule Network and a 11.49% reduction in the model built by DC++Net.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. *"ImageNet Classification with Deep Convolutional Neural Networks."* 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *"Deep Residual Learning for Image Recognition."* In Proceedings of the IEEE Conference on Computer Vision and Patter Recognition. arXiv preprint arXiv: 1512.03385, 2015. [online] [https://arxiv.org/pdf/1512.03385.pdf] [Accessed on July 20, 2019].

[3] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. *"Dynamic routing between capsules."* In Advances in Neural Information Processing Systems, pages 38593869, 2017.

[4] Rupesh Kumar Srivastava, Klaus Greff, and Jrgen Schmidhuber. *"Highway networks."* CoRR, abs/1505.00387, 2015.

[5] Geoffrey E Hinton, Sara Sabour, Nicholas Frosst. *"Matrix Capsules with EM Routing."* 2018. [online] [https://openreview.net/pdf?id=HJWLfGWRb] [Accessed on July 20, 2019].

[6] Edgar Xi, Selina Bing, Yang Jin. *"Capsule Network Performance on Complex Data."* arXiv preprint arXiv:1712.03480, 2017. [online] [https://arxiv.org/pdf/1712.03480 .pdf] [Accessed on July 20, 2019].

[7] Geoffrey E Hinton, Alex Krizhevsky, Sida D Wang. *"Transforming Auto-Encoders."* In International Conference on Arti
cial Neural Networks. Springer, 2011.

[8] Sai Samarth R Phaye, Apoorva Sikka, Abhinav Dhall, Deepti Bathula. *"DCNET and DCNET++: Making the Capsules Learn Better."* arXiv:1805.04001, 2018. [online] [https://arxiv.org/pdf/1805.04001.pdf] [Accessed on July 20, 2019].

[9] Venkitachalam, S., Manghat, S.K., Gaikwad, A.S., Ravi, N., Bhamidi, S., El-Sharkawy, M., *"Realtime Applications with RTMaps and Bluebox 2.0"*. International Conference Artificial Intelligence 2018 ICAI'18.

[10] SSRP, Multi-level-DCNet.[online] [https://github.com/ssrp/Multi-level-DCNet] [Accessed on July 20, 2019].

[11] Yann LeCun, Corinna Cortes, Christopher JC Burges. *"The MNIST Database of Handwritten Digits,"* 1998. [online] [http://yann.lecun.com/exdb/mnist/] [Accessed on July 20, 2019].

[12] Tom Hope, Yehezkel S. Reshe, Itay Lieder (2017-08-09). *"Learning TensorFlow: A Guide to Building Deep Learning Systems."* "O'Reilly Media, Inc. pp. 64. ISBN 9781491978481.

[13] Jonathan Hui. *"Understanding Dynamic Routing between Capsules (Capsule Networks)."* [online] [https://jhui.github.io/2017/11/03/Dynamic-Routing-Between-Capsules/] [Accessed on July 20, 2019].

[14] Yann LeCun, Leon Bottou, Yoshua Bengio, Patrick Haffner. *"Gradient-Based Learning Applied to Document Recognition."* Proceedings of the IEEE, vol. 86, no. 11, 1998, pp. 2278-2324., doi:10.1109/5.726791. [online] [https://ieeexplore.ieee.org/document/726791] [Accessed on July 20, 2019].